

GAPLE: Generalizable Approaching Policy LEarning for Robotic Object Searching in Indoor Environment

Xin Ye¹, Zhe Lin², Joon-Young Lee², Jianming Zhang², Shibin Zheng¹ and Yezhou Yang¹

Abstract—We study the problem of learning a generalizable action policy for an intelligent agent to actively approach an object of interest in an indoor environment solely from its visual inputs. While scene-driven or recognition-driven visual navigation has been widely studied, prior efforts suffer severely from the limited generalization capability. In this paper, we first argue the object searching task is environment dependent while the approaching ability is general. To learn a generalizable approaching policy, we present a novel solution dubbed as GAPLE which adopts two channels of visual features: depth and semantic segmentation, as the inputs to the policy learning module. The empirical studies conducted on the House3D dataset as well as on a physical platform in a real world scenario validate our hypothesis, and we further provide in-depth qualitative analysis.

I. INTRODUCTION

Enabling an autonomous robot to search and retrieve a desired object in an arbitrary indoor environment is always both fascinating and extremely challenging, as it would enable a variety of applications that could improve the quality of human life. For example, being able to navigate and localize objects is one of the basic functions that a robot elderly caregiver should be equipped with. Such technology can also be potentially used to help visually impaired people, thus significantly improving their quality of life. Moreover, self-driving cars with such an object searching capability will be able to approach and pick up their designated customers. Fundamentally, having a robot with vision that finds object is one of the major challenges that remain unsolved.

With the current surge of deep reinforcement learning [1]–[3], a joint learning method of visual recognition and planning emerges as end-to-end learning [4], [5]. Specifically, the robot learns an optimal action policy to reach the goal state by maximizing the reward it receives from the environment. Under the “robot that finds objects” setting, the goal state is the location of the target object with a high reward assigned. Several recent work have attempted to fulfill the challenge and achieved certain promising results. [4] adopted a target-driven deep reinforcement learning model to let robot find a specific visual scene. [5] also proposed a recognition-guided deep reinforcement learning for robot to find a user-specified target object. Although these deep reinforcement learning models can be trained to navigate a robot to find a target scene or object in an environment,

a time-consuming re-training process is needed every time the target or the environment alters. In other words, these systems suffer from an unsatisfiable generalization capability to transfer the previously learned action policy to a brand new target or a novel environment. Such defect extremely limits the applications of these methods in real-world scenarios as it is impractical to conduct the inefficient training process every single time. In this paper, we argue that the limitation is deeply rooted in the task itself. While searching an object is indeed environment and object dependent, approaching an object after seen once should be a general capability. The insight could also be explained while observing human beings searching an object in a house. We first need to explore the house to locate the object once. After the object is captured with one sight, we are able to approach the target object with fairly few back and forth explorations. While the exploration policy varies a lot, the optimal approaching policy is indispensable, and provide a critical last step for a successful object search. Thus approaching policy is a much general capability of human beings, and thus in this paper, we focus on the approaching policy learning. We define an approaching task as the robot is initialized in a state where the target object can be seen, and the goal is to take the minimal number of steps to approach the target object.

To tackle the challenge, we put forward a novel approach aiming at learning a generalizable approaching policy. We first treat a deep neural network as the policy approximator to map from visual signals to navigation actions, and adopt the deep reinforcement learning paradigm for model training. The trained model is expected to navigate the robot approaching a new target object in a new environment without any extra training effort. To learn an optimal action policy that can lead to a shortest path to approach the target object, previous methods typically attempts to map visual signal to an optimal action directly, no matter the signal contains clues towards reaching the goal state or not. In such a case, these methods inherently force the policy network to encode the local map information of the environment, which is specific towards a certain scene. Thus, re-training or fine-tuning is needed to update the model parameters while facing a novel target object or a new environment.

Rather than learning a mapping from each visual signal directly to a navigation action, which has a much higher chance of encoding environment-dependent features, we present a method that first explicitly learns a general feature representations (scene depth and semantic segmentation map) to capture the task-relevant features solely from the visual signal. The representations serve as the input to the deep

¹ X. Ye, S. Zheng and Y. Yang are with the Active Perception Group at the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA, Email: {xinye1, szheng31, yz.yang}@asu.edu

² Z. Lin, J. Lee and J. Zhang are with Adobe Systems, Inc. San Jose, CA, USA, Email: {zlin, jolee, jianmzha}@adobe.com

reinforcement learning model for training the action policy. To validate our proposed method’s ability to generalize the approaching behavior, empirical experiments are conducted on both simulator (House3D) and in a real-world scenario. We report the experimental results (a sharp increase of the generalization ability over baseline methods) in Section IV.

II. RELATED WORK

Target-driven visual navigation. Among plenty of methods for target-driven visual navigation, those ones with deep reinforcement learning are most relevant, as we will not provide any human guidance or map related information to the learning system. Recently, [6] approached the target-driven deep reinforcement learning problem by jointly conducting depth prediction with other classification tasks. [4] proposed a target-driven framework to enable a robot to reach an image-specified target scene. [5] introduced a recognition-guided paradigm for robot to find a target object in indoor environments. Both [7] and [8] aim to let robot navigate in an indoor environment and collect necessary information to answer a question. Although these methods work well in their designed domain, the generalization ability towards new object and new environment is questionable. More recently, [9], [10] attempted to benchmark and study a variety of visual representation combinations for target-driven navigation, showed that using the segmentation and detection mask yields a higher generalization ability.

Generalization in deep reinforcement learning. While generalization ability is a critical evaluation criteria in deep learning, it is less mentioned in the literature of deep reinforcement learning, where most of work focus on improving the training efficiency and the performance of the trained model in certain specific domains [1]–[3], [11]–[14]. The authors of [15] proposed to predict the effect of different actions on future measurements, resulting in good generalization ability across environments and targets. However, it is based on the condition of training the model in the complex multi-texture environments. [16] adopted the error in predicting the consequences of robot’s actions as curiosity to encourage robot to explore the environment more efficiently. Yet it still needs a fine-tuning process when deploying the trained policy in a new environment. [17] addressed the driving policy transfer problem by means of modular design and abstraction, but they only studied the problem w.r.t supervised deep learning methods, without considering the more challenging generalization issue in deep reinforcement policy learning.

Semantic segmentation and depth prediction. Semantic segmentation and depth prediction from a single image are two fundamental tasks in computer vision and have been extensively studied. Recently, convolutional neural network and deep learning based methods show dominating performance to both tasks [18]–[22]. Instead of addressing them separately, [23] proposed a unified framework for jointly predicting semantic and depth from a single image. [24] also adopted a single neural network to do semantic labeling, depth prediction and surface normal estimation. In work

[25], the authors analyzed the cross-modality influences between semantic segmentation and depth prediction and then designed a network architecture to balance the cross-modality influences and achieve improved results. Despite the good performance these methods achieved, multi-step training process is still required, that leads to heavy computational load in learning and using these models. In this paper, we adopt a *DeepLabv3+* [21] based model with which we can perform end-to-end training without a performance loss.

III. OUR APPROACH

A. Overview

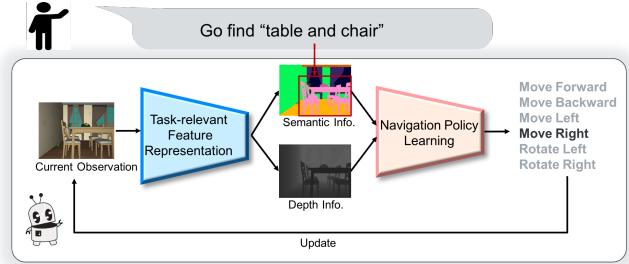


Fig. 1: An overview of our GAPLE system.

We define the task as learning a generalizable action policy for a robot to approach a user-specified target object with minimal steps. The target object is specified as its semantic category. The robot’s on-board camera is the only sensor to capture RGB images, which serve as the robot’s observations. The robot starts at the location where the target object can be detected. With the current observation, the robot makes a decision upon which action to take. Afterwards, the robot receives a new observation and it repeats the decision process iteratively until it reaches a close enough location to the target object. Moreover, once the action policy is trained and deployed, the robot is expected to take reasonable number of steps to approach the user-specified target object as soon as the robot sees it, even the target object is from a new category or the environment changes.

Fig. 1 shows an overview of our system. Since the action decision depends on the robot’s current observation, the RGB image is the input to the system. Besides, to make the system be flexible to the appearance changes of the target object in the same semantic category, we further include its semantic label as part of the input.

To generalize well across various environments, the feature representations from the input image should be also general across all different environments, or so-called environment-independent. Although the deep neural network is well-suited for extracting task-relevant features [26], it tends to capture the environment-dependent features. For example, [4] also pointed out that a scene-specific layer is needed to capture the special characteristics like the room layouts. As a result, these models that integrally learns feature representation and navigation policies, can be easily over-fitted towards specific environments. To overcome this challenge, we propose

to explicitly learn a more general feature representations. Consider our object approaching task as an example, the robot needs to capture the semantic information from its observation to identify the target object. At the same time, the depth information is crucial for the robot to navigate and avoid collisions. Thus, we adopt a feature representation module that captures both the semantic and the depth information from the input image. We further pipeline the outputs of our feature representation module as the inputs to our proposed navigation policy learning module for action policy training. The following sections introduce the feature representation module and the navigation policy learning module respectively.

B. Semantic Segmentation and Depth Prediction

We adopt the *DeepLabv3+* [21] based model (as shown in Fig. 2) to jointly predict the semantic segmentation and depth map from a single RGB image. *DeepLabv3+* employs an encoder-decoder structure, where the encoder utilizes the spatial pyramid pooling to encode multi-scale contextual information. Specifically, it applies multiple filters or pooling operations that with different rates on the feature map computed by other pretrained models, such as ResNet-101 [27] and Xception [28]. That allows the filters or the pooling operations to be able to consider different field-of-views, so that they can capture rich semantic information. During the decoding process, it gradually up-samples the encoder’s output and recovers the spatial information to capture the sharp object boundaries, which leads to better semantic segmentation results. We refer interested readers to [21] for more details.

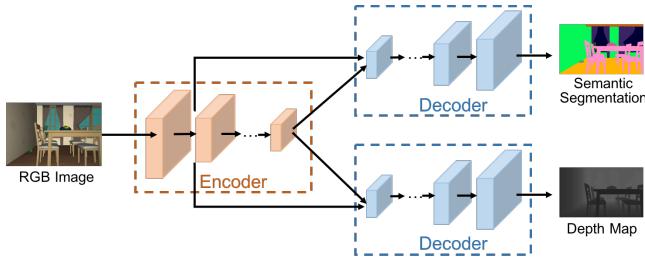


Fig. 2: An illustration of the adopted model based on *DeepLabv3+* [21] to predict semantic segmentation and depth map from a single RGB image.

For generating the depth map at the same time, we spawn another decoder branch. The motivation is that the depth information and semantic segmentation are correlated. Either one can be used as a guidance to help predicting the other one according to [18], [19]. Thus it is benefiting to jointly predict both of them [23], [24]. Here, we adopt the exactly same architecture as the one for semantic segmentation except for the output layer. Rather than a classification layer that outputs labels for each corresponding pixel, we utilize a regression layer instead to predict depth value for each pixel.

$$L = \frac{1}{N} \sum_i^N (-p_i^* \log(p_i)) + \lambda \frac{1}{N} \sum_i^N \|d_i - d_i^*\|_2^2 \quad (1)$$

Specifically, our system adopts the Xception-65 model [28] pretrained on ImageNet [29] as initialization. We then define the loss function (Eq. 1) to train our model in an end-to-end manner. The first term is the cross entropy loss for semantic segmentation. p_i^* is the one-hot encoded ground-truth semantic label for pixel i . p_i is the corresponding predicted probabilities over all possible semantic labels. The second term is the mean-square error for depth prediction, where d_i^* denotes the ground truth depth value for pixel i and d_i represents the corresponding predicted depth value. N denotes the total number of pixels in the image and λ denotes a balancing factor. In practice, $\lambda = 0.01$ achieves good performance empirically and we train our model by minimizing the loss function through the stochastic gradient decent (SGD) optimization.

C. Approaching Policy Learning

With the semantic segmentation and the depth information computed as the representations of the robot’s current observation, the robot is expected to make a decision of which action to take to approach the target object. Consider the challenge that the overall **state space** for robot is unknown and each state is of high dimension, we apply the deep reinforcement learning method.

First, we design a deep neural network as an estimator of the policy function. The **policy network** takes both semantic segmentation and depth information as inputs and outputs a probability distribution over all **valid actions** (also known as action policy). The robot picks a valid action either randomly or follows the distribution predicted by the policy network. After performing the action, the robot receives a **reward signal** as a measurement of how beneficial the performed action is towards the goal. This one-step reward (or likewise the accumulated rewards after taking multiple steps) serves as the weight factor of taking the performed action as the ground truth action for training. We further introduce each part of our setting in details here.

State space: Since we assume that the RGB image captured by robot’s camera is the only source of information, and both of the robot’s position and the target object’s location are unknown, the robot’s state can only be represented by the captured RGB image as well as the semantic label of the target object. As mentioned before, we represent the RGB image using semantic segmentation and depth map, and the semantic segmentation together with the semantic label of the target object can be further encoded as an attention mask. Afterwards, the attention mask and the depth map together represent the robot’s state (see left side of Fig. 3). In addition, the size of the attention field also encodes how close the robot to the target object. Thus, we set a threshold and set the **goal states** as those with an attention field larger than the threshold. In practice, we set it as the size of fifth largest attention field among all ground truth attention masks to yield five goal states. All the possible states form the state space.

Action space: To constrain the number of possible states, we only consider discrete actions. Without loss of generality, we consider some basic actions for the navigation purpose,

namely “move forward”, “backward”, “left or right with a fixed distance”, “rotate left or right with a constant angle”. In our experiments, we define a fixed distance as 0.2 meters and a constant angle as 90 degrees.

Reward function: We adopt the reward function designed by [5] to avoid getting stuck in certain suboptimal states. We define the reward as the size of the attention field if and only if the attention field is larger than all previous ones the robot has observed. Otherwise, the reward is set to be zero. Formally, let a_t be the size of the attention field the robot observes at time step t , the reward at this time step $r_t = a_t$ if and only if $a_t > a_{t-1}, a_{t-2}, \dots, a_0$, otherwise $r_t = 0$. As a result, the discounted cumulative reward for one episode will be $\gamma^{i_1} a_{i_1} + \gamma^{i_2} a_{i_2} + \dots + \gamma^{i_t} a_{i_t}$, where γ is the discount factor for time penalty and $a_{i_1} < a_{i_2} < \dots < a_{i_t} (i_1 < i_2 < \dots < i_t)$.

Policy network: Fig. 3 illustrates the overall policy learning architecture. The learning module takes the semantic segmentation and depth map as inputs. The semantic segmentation is then used to create an attention mask with the semantic label of the target object. We further resize both the attention mask and the depth map to the size of 10 by 10, and then concatenate them into a joint vector before attaching a fully connected layer to generate an embedding fusion. The embedding fusion is then feed into two separate branches, each of which consists of two additional fully connected layers to predict action policy and the state value respectively, where the state value is defined as the expected cumulative reward the robot would receive at the current state.

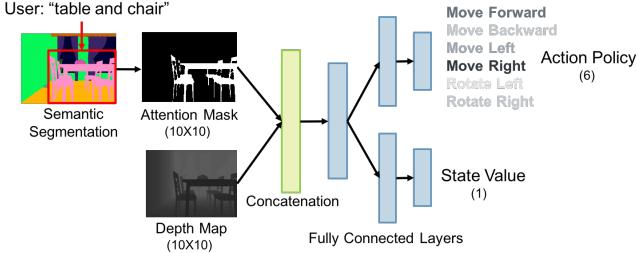


Fig. 3: The architecture of our deep reinforcement learning model for action policy training.

We follow the training protocol from [4]. It trains the model by running multiple threads in parallel and each thread updates the weights of the global shared network asynchronously. However, rather than assigning each thread a specific environment-target pair, we adopt a scheduler with a work stealing manner ([30]) in order to train all environment-target pairs equally, in case of certain environment-target pairs are much easier to train.

IV. EXPERIMENTS

A. Dataset

To train our model and test its generalization ability across different target objects and environments, we need a data efficient platform that has a diverse set of objects and environment types. Here, we adopt the publicly available

simulation platform House3D [31], a renderer that builds on SUNCG dataset [32]. Because House3D consists of rich 3D indoor environments for a virtual robot to interact with. During the interaction, the robot has access to the first-person view RGB images, as well as the corresponding ground truth semantic segmentations and depth maps, which makes it well suited to the feature representation learning task and the approaching policy learning task. Fig. 4 (a) depicts an example data.

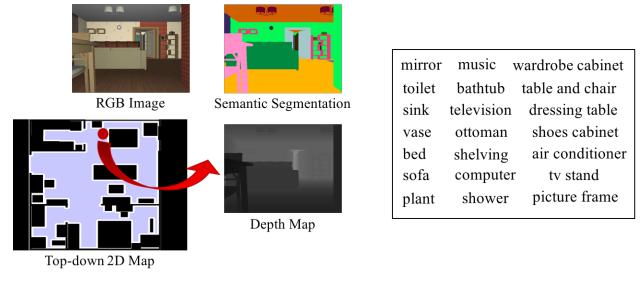


Fig. 4: A sample environment from House3D and some target object candidates.

We constrain the robot to perform discrete actions in these virtual environments, i.e. moving 0.2 meters or rotating 90 degrees every time. It also discretizes the environment into a set of reachable locations. We select a total of 248 simulated environments that are suitable for testing. Additionally, to avoid ambiguity, we select the objects that only have one instance in an environment as the target objects for robot to approach. Fig. 4 (b) lists example target objects used.

B. Semantic Segmentation and Depth Prediction

In order to train our feature representation module for semantic segmentation and depth prediction, we collect RGB images, as well as their corresponding ground truth captured at all discrete locations from 100 environments. We further delete the images that has over 80% background and randomly sample a total of 55,697 images for training. For semantic segmentation, 77 semantic labels are of our interest, with all the remaining ones being classified as “background”.

We take the popularly used metrics, a.k.a. mean Intersection Over Union (mean IOU) and Root Mean Square Error (RMSE) to report the performance of our trained models in doing semantic segmentation and depth prediction respectively. Our model achieves 0.436 in mean IOU for semantic segmentation on validation dataset, and 0.0003 normalized RMSE for depth prediction. Fig. 5 shows several qualitative results.

C. Approaching Policy Learning

To demonstrate the generalization ability of our proposed method across both target objects and environments, we compare our method with the following baselines and variants. Again, the map of the environment is unknown to all



Fig. 5: Some qualitative semantic segmentations and depth predictions from our feature representation module.

methods, except when calculating the minimal steps that robot needs to take to approach the target object.

a) Random method. At each state, the robot randomly choose an action to perform. Since the map is unknown, the action might yield collision. In that case, the robot will simply stuck in the current state. The random method provides a performance lower-limit, which could calibrate how “intelligent” the other trained models are.

b) Method from [5]. It takes the output from the *res4* layer of ResNet-50 network that pretrained on ImageNet as the feature representation from both the target object and the robot’s current observation. The two channels of feature representations, as well as a binary attention mask that is generated using an object recognition module form the inputs to the deep reinforcement learning model. Here, we first use the ground truth attention mask to remove the influence from the noisy object recognition module. Then we test the method with the attention mask generated from our predicted semantic segmentation. Moreover, we adopt a single scene-specific branch for all the target objects and environments.

c) Our method with ground truth semantic segmentation and depth map. We take the ground truth semantic segmentation and depth map as the inputs to our deep reinforcement learning model as described in Sec. III-C, for the purpose of testing the performance upper-limit.

d) Our method with ground truth semantic segmentation and predicted depth map. This method is used to compare with the method b) as they both adopt ground truth semantic info to generate noise-free attention mask.

e) Our proposed method described in Sec. III that takes only an RGB image and the semantic label of the target object as the inputs, and outputs navigation actions.

We train and evaluate all the methods under two settings for object-wise generalization and environment-wise generalization respectively. **Setting 1:** training on 6 different target objects in 1 environment. The models trained on this setting are then used to evaluate their generalization abilities across target objects. To be specific, during the testing, we use the trained models to approach 5 unseen target objects in the same environment and report their performances respectively. **Setting 2:** training on a total of 24 target objects in 4 different environments (6 each). We then test the trained models’ performances in approaching another 24 target objects in 4 novel and unseen environments. For both settings, the robot always starts at the position where the target object can be detected for both the training and the testing phases.

To conduct fair comparisons, for each testing environment-object pair, we randomly select 100 starting positions. The robot stops either it reaches close enough to the target object (a successful case) or it reaches 1000 steps (a failure case). We take two metrics to compare these methods, namely the success rate in terms of how many steps (relative to the minimal steps) are taken, and the average steps over the successful cases. Generally speaking, a higher success rate or a smaller number of average steps indicates a better approaching performance.

We trained each of these models with an Nvidia V100 (6 cards with 16g memory each) machine. For setting 1, each model’s training takes about 20 hours. For setting 2, the training takes about 40 hours to converge.

Fig. 6, Table I and Table II report the achieved success rate, success rate drop from trained objects/environments to new objects/environments, and average steps of all methods on the two settings respectively. From Fig. 6 and Table I, the results indicate a clear generalization capability improvement of our method d) comparing with the method b) that both take ground truth attention mask, and our method e) with the method b) that both take predicted attention mask. At the same time, the results align well with our expectation that our method with the predicted semantic segmentation (method e)) performs worse than the method d), which also happens for method b). The reason is due to the recognition errors introduced from the predicted semantic segmentation that distracts the robot from approaching the target object. More specifically, the reward generated upon the area of the target object is not consistent due to the noisy detection. Moreover, the area of the robot’s attention (focusing on the target object) also needs to encode the goal states. With the noisy predicted semantic labels, the robot has a high likelihood to get stuck while it struggles to identify the correct goal states.

Table II reports the average number of steps taken among all successful trials. With the success rate reported in Fig. 6, it also matches our expectation that the average number of steps from our methods are generally larger than the method b). Here, the larger number of steps means that our methods also succeed in approaching the target object which needs larger number of steps, while the method b) fails these cases and they don’t contribute to the average number of steps.

D. Real World Experiment

We adopt our method (method d)) in a real world scenario (on a public dataset from [5]). Without further fine-tuning the trained model, our trained model can still guide the robot to

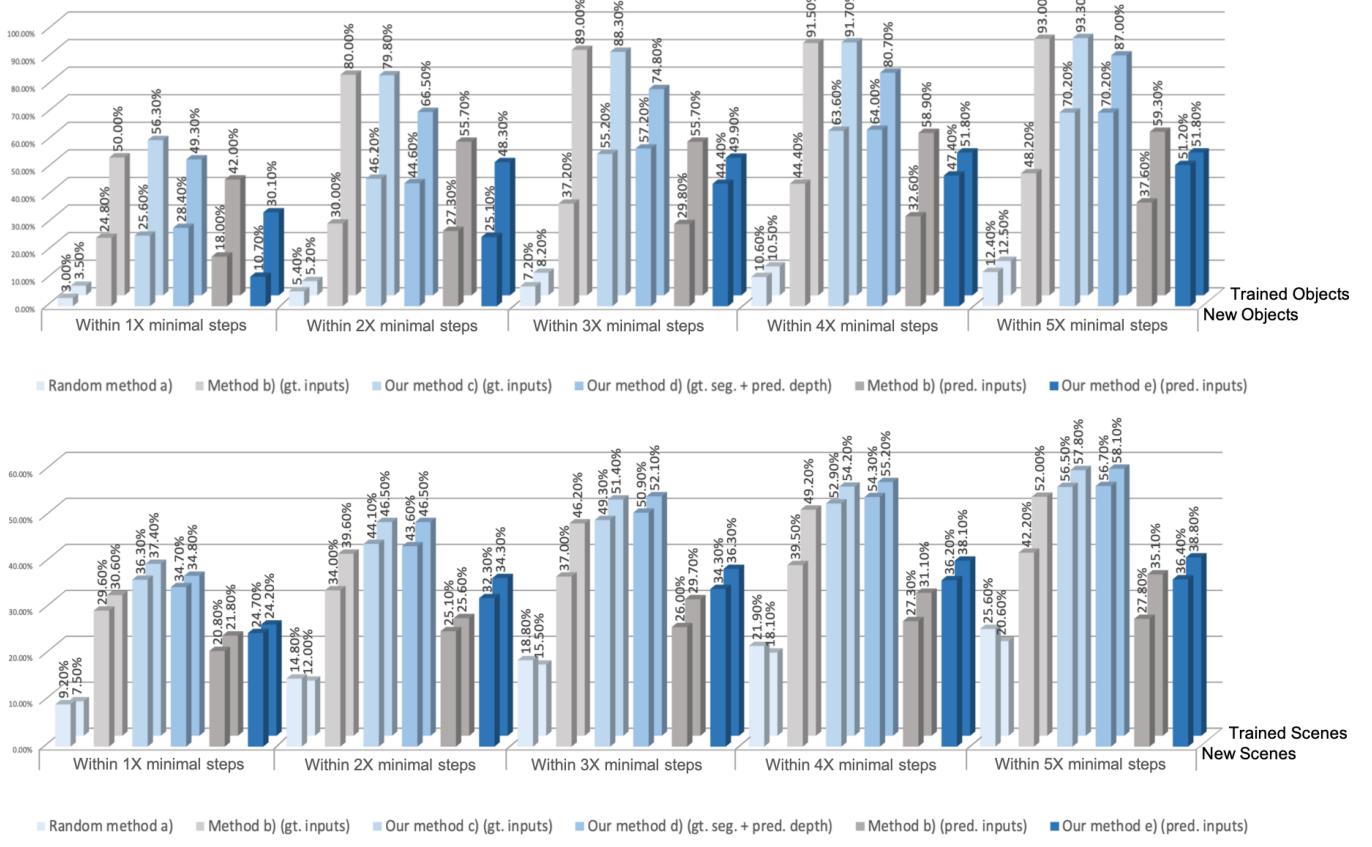


Fig. 6: Successful approaching rates. Upper: Setting 1: generalization ability across target objects (on trained objects and on new objects). Lower: Setting 2: generalization ability across environments (on trained environments and on new environments).

TABLE I: The success rate drop from the trained objects to new objects (setting 1: s1), and from the trained environments to new environments (setting 2: s2).

Δ success rate	1X minimal steps		2X minimal steps		3X minimal steps		4X minimal steps		5X minimal steps	
	s1	s2								
Random method a)	0.50%	-1.70%	-0.20%	-2.80%	1.00%	-3.30%	-0.10%	-3.80%	0.10%	-5.00%
Method b) (gt. inputs)	25.20%	1.00%	50.00%	5.60%	51.80%	9.20%	47.10%	9.70%	44.80%	9.80%
Our method c) (gt. inputs)	30.70%	1.10%	33.60%	2.40%	33.10%	2.10%	28.10%	1.30%	23.10%	1.30%
Our method d) (gt. seg. + pred. depth)	20.90%	0.10%	21.90%	2.90%	17.60%	1.2%	16.70%	0.90%	16.80%	1.40%
Method b) (pred. inputs)	24.00%	1.00%	28.40%	0.50%	25.90%	3.70%	26.30%	3.80%	21.70%	7.30%
Our method e) (pred. inputs)	19.4%	-0.50%	23.20%	2.00%	5.50%	2.00%	4.40%	1.90%	0.60%	2.40%

approach the target object. For this real world experiment, we use the trained model from [20] to predict depth map and the ground truth bounding box to generate the attention mask. Fig. 7 shows an example of how the robot approaches the target object, which is a “whiteboard”.

E. Analysis and Discussion

To better understand why our proposed method achieves better generalization ability, we further conduct analysis from the feature representation perspective. Generally speaking, the goal of the deep reinforcement learning is to learn a policy model π that maps a state s to the most “beneficial” action a (or an action distribution from which the most

“beneficial” action can be drawn with a high probability), i.e. to let $\pi(s) = a$. This most “beneficial” action a , unlike the ground truth label in a general supervised learning problem, is acquired by the intelligent agent’s trial and error interactions with the environment.

For our object approaching task, the most “beneficial” action a essentially depends on the local map between the current location and the goal location. In order to let $\pi(s) = a$, if the input s doesn’t provide any map information directly (such as the setting from [5]), then the model π has to capture such information from the input s through learning. To avoid the over-fitting problem, it is necessary to train the model π on a large enough and diverse enough training data where

TABLE II: Average number of steps taken by all methods on two settings.

methods	setting (1)		setting (2)	
	trained obj.	new obj.	trained env.	new env.
minimal	3.88	3.89	2.58	2.06
Random a)	213.71	202.15	166.62	109.85
Method b) gt.	5.88	7.85	4.34	3.64
Method b) pred.	4.70	5.82	6.15	5.94
Our method c)	5.85	13.99	6.14	5.14
Our method d)	8.45	13.43	7.96	5.96
Our method e)	10.31	16.44	3.77	4.22

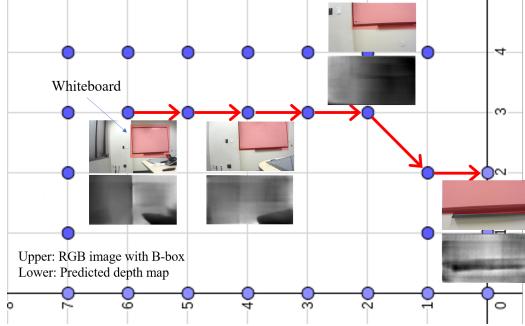


Fig. 7: An example of the mobile robot approaches the target object “whiteboard” using the method (d)). Upper view: RGB input; Lower view: Depth map generated.

the underlying distribution of the relations between the state s and the map information can be captured. Though it is a straightforward, the well-known sample-inefficient issue lingering in the paradigm of deep reinforcement learning makes it fairly impracticable.

In this work, we first adopt a feature representation model f to learn the semantic segmentation and depth map from the input state s , then we take the semantic segmentation and the depth map as the inputs to the policy model π . In other words, we aim to let $\pi(f(s)) = a$. Here, we hypothesize that the depth map as an input to the policy network π encodes the local map well already. In such a way, the policy model π is not the only source for capturing the local map information well. At the same time, the feature representation model f directly learns the depth map from the state s in a supervised manner, which is much more sample efficient.

For further validation, we examine the relationship between the distance in physical space and the one in the feature space. For each pair of the locations in an environment, we calculate their Manhattan distance in terms of steps as the physical distance. We adopt L_1 distance between the normalized feature maps of the images taken at the two locations with the same orientation as the feature distance. Fig. 8 shows the relations between the physical distance and the distance in both depth feature space and ResNet-50 feature space.

From Fig. 8, it shows that within a small range of physical distances (1 to 9 steps), the distance in depth feature space increases notably along the increment of the physical distance. While the physical distance is out of this range (over 9 steps), the feature distance shows minor changes. This observation suggests that depth feature captures the differences between

different locations within a small region, which aligns well with our assumption. On the other hand, the distance in ResNet-50 feature space grows almost independently w.r.t. the growing of the physical distance. We speculate that this observation provides the actual reason why methods (such as [5]) fails to generalize well.

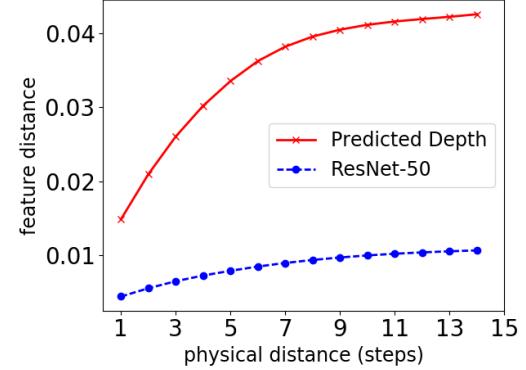


Fig. 8: Pair-wise feature distances w.r.t. physical distances.

V. CONCLUSION

This paper presents a novel approaching policy training paradigm dubbed as GAPLE, through explicit depth estimation and semantic segmentation. Empirical studies on the House3D platform and a real physical experiment on a mobile robot validate that the new framework is able to yield a significantly higher generalization capability towards new target objects and novel environments, indicating a promising pathway for future research on achieving generalizable object searching policy on mobile robots.

REFERENCES

- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.
- [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.
- [3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*, 2016, pp. 1928–1937.
- [4] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, “Target-driven visual navigation in indoor scenes using deep reinforcement learning,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3357–3364.
- [5] X. Ye, Z. Lin, H. Li, S. Zheng, and Y. Yang, “Active object perceiver: Recognition-guided policy learning for object searching on mobile robots,” in *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*. IEEE, 2018.
- [6] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, “Learning to navigate in complex environments,” *arXiv preprint arXiv:1611.03673*, 2016.
- [7] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, “Embodied question answering,” *Proceedings of CVPR, Salt Lake City, Utah, USA*, 2018.
- [8] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, “Iqa: Visual question answering in interactive environments,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4089–4098.

[9] P. Ammirato, P. Poirson, E. Park, J. Košeká, and A. C. Berg, “A dataset for developing and benchmarking active vision,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1378–1385.

[10] A. Mousavian, A. Toshev, M. Fiser, J. Kosecka, and J. Davidson, “Visual representations for semantic target driven navigation,” *arXiv preprint arXiv:1805.06066*, 2018.

[11] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu, “Reinforcement learning with unsupervised auxiliary tasks,” *arXiv preprint arXiv:1611.05397*, 2016.

[12] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, “Sample efficient actor-critic with experience replay,” *arXiv preprint arXiv:1611.01224*, 2016.

[13] D. Ghosh, A. Singh, A. Rajeswaran, V. Kumar, and S. Levine, “Divide-and-conquer reinforcement learning,” *arXiv preprint arXiv:1711.09874*, 2017.

[14] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3389–3396.

[15] A. Dosovitskiy and V. Koltun, “Learning to act by predicting the future,” *arXiv preprint arXiv:1611.01779*, 2016.

[16] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, “Curiosity-driven exploration by self-supervised prediction,” in *International Conference on Machine Learning (ICML)*, vol. 2017, 2017.

[17] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun, “Driving policy transfer via modularity and abstraction,” *arXiv preprint arXiv:1804.09364*, 2018.

[18] C. Liu *et al.*, “Beyond pixels: exploring new representations and applications for motion analysis,” Ph.D. dissertation, MIT, 2009.

[19] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, “Indoor semantic segmentation using depth information,” *arXiv preprint arXiv:1301.3572*, 2013.

[20] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 239–248.

[21] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *arXiv preprint arXiv:1802.02611*, 2018.

[22] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011.

[23] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, “Towards unified depth and semantic prediction from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2800–2809.

[24] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[25] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother, “Analyzing modular cnn architectures for joint depth prediction and semantic segmentation,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4620–4627.

[26] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*, 2014, pp. 647–655.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[28] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” *arXiv preprint*, pp. 1610–02357, 2017.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[30] S. Chen, P. B. Gibbons, M. Kozuch, V. Liaskovitis, A. Ailamaki, G. E. Blelloch, B. Falsafi, L. Fix, N. Hardavellas, T. C. Mowry *et al.*, “Scheduling threads for constructive cache sharing on cmps,” in *Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures*. ACM, 2007, pp. 105–115.

[31] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, “Building generalizable agents with a realistic and rich 3d environment,” *arXiv preprint arXiv:1801.02209*, 2018.

[32] D. Song, N. Kyriazis, I. Oikonomidis, C. Papazov, A. Argyros, D. Burschka, and D. Kragic, “Predicting human intention in visual observations of hand/object interactions,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1608–1615.